T.D

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/674,149	09/29/2003	Ajay K. Gupta	SVL920030078US1	1915

43166 7590 06/14/2006

DRIGGS, HOGG & FRY CO., L.P.A.
38500 CHARDON ROAD
DEPT. ISV
WILLOUGBY HILLS, OH 44094

EXAMINER	
VAUTROT, DENNIS L	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 06/14/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/674,149	GUPTA, AJAY K.	
	Examiner	Art Unit	
	Dennis L. Vautrot	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 September 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *9/29/2003*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Information Disclosure Statement

1. The information disclosure statement (IDS) submitted on 29 September 2003 has been received and entered into the record. Since the IDS complies with the provisions of MPEP § 609, the references cited therein have been considered by the examiner. See attached forms PTO-1449.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claim 14 is rejected under 35 U.S.C. 102(b) as being anticipated by **Holenstein et al.** (US 2003/0037029). **Holenstein et al.** teaches an article of manufacture comprising one or more non-volatile storage media encoding instructions for performing a high availability data replication process that synchronizes a secondary server with a primary server of a database that includes smart large objects (See page 2, paragraph [0029] "Plural databases are synchronized in a database replication system that includes a plurality of nodes connected via communication media in a topology..." and

page 8, paragraph [0144] "By inserting the marker at the appropriate point in the audit trail, the BLOB can be sent via a separate channel (or perhaps read directly by the CONS) that is optimized for BLOB access..."), the process comprising: ordering log entries of a smart large object modifying operation performed on the primary server in a selected order wherein a log entry corresponding to updating a large object header of said smart large object is consistent immediately upon execution (See page 5, paragraph [0084] "...and a transaction transmitter which sends selected transactions in the audit trail to one or more other nodes." These transactions are modifying operations performed on the primary server. And See page 5, paragraph [0087]. "The marker is written to the audit trail at the first node." It is interpreted that included in the "audit trail" is the header as used in the reference.); transferring log entries including said log entries of said smart large object modifying operation from the primary server to the secondary server (See page 5, paragraph [0084] "...and a transaction transmitter which sends selected transactions in the audit trail to one or more other nodes..."); and replaying said transferred log entries on the secondary server, the replaying of said log entries of said smart large object modifying operation being performed in the selected order without locking said smart large object on the secondary (See page 5, paragraphs [0089-0090] "The block of data is sent from the first node to the second node without passing the block of data through the audit trail. At the second node, the block of data is stored in the target database upon receipt at the second node of the block of data and the marker assigned to the block of data. An insert operation may be used to perform the function." Avoiding the audit trail keeps the smart large object from having to be

locked on the secondary.)

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-6, and 8 - 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Bruso et al.** (6,615,219) in view of **Holenstein et al.** (US 2003/0037029).

6. Regarding claim 1, **Bruso et al.** teaches a method comprising: initiating a transaction on the secondary server that includes reading a smart large object (See column 8, lines 48-53 "If the referenced BLOB is cached, BLOB handler reads the BLOB from the cache and returns the BLOB to Web BLOB server, which returns the BLOB to the requesting application. Otherwise, BLOB handler uses the URL to read the BLOB from BLOB storage."); responsive to the initiating, creating a memory cache corresponding to the smart large object (See column 8, lines 62-64 "At step 402, DBMS allocates storage for the row of data to insert, and BLOB storage is allocated at step 404."), said memory cache including a large object header (See column 8, lines 66-67

"At step 406, the BLOB header is initialized with the information previously described."); and responsive to a replay on the secondary server of a log entry by said replicator that alters said smart large object, sending an exception to said transaction (See column 9, lines 19-24 "...if the DBMS detects a processing error which prevents the insert-row operation from being committed, the BLOB pages allocated at step 404 are returned to the appropriate free chain of the allocation control table, along with the BLOB header page.")

Bruso et al. fails to teach a database including primary and secondary servers and a replicator that copies database log entries between the primary server and the secondary server and replays said log entries on the secondary server.

However **Holenstein et al.** teaches a database including primary and secondary servers and a replicator that copies database log entries between the primary server and the secondary server (See page 2, paragraph [0029] "The transaction transmitter at the first node send the marker in the audit trail to a second node having a target database." The "audit trail" of the reference is defined to be the same as the database log of the claim.) and replays said log entries on the secondary server (See page 2, paragraph [0029] "At the second node, the block of data is stored in the target database upon receipt at the second node of the block of data and the marker assigned to the block of data.")

It would have been obvious to one with ordinary skill in the art to combine the method of processing large objects as disclosed in **Bruso et al.** with the replication system of databases as disclosed in **Holenstein et al.** because it is common to store large objects on a database and databases are frequently replicated. As disclosed in the specification on page 2, handling large objects can cause compatibility issues with database replication and by combining **Bruso et al.'s** method to process large objects with the more generic database replication disclosure of **Holenstein et al.** the large object replication problem is addressed. It is for this reason that one of ordinary skill in the art would have been motivated to include a database including primary and secondary servers and a replicator that copies database log entries between the primary server and the secondary and replays said log entries on the secondary server.

7. Regarding claim 2, **Bruso et al.** additionally teaches committing said replay of said log entry on the secondary server to alter said smart large object on the secondary server (See column 5, lines 39-41); updating said large object header to generate an updated large object header (See column 6, lines 24-36); and completing said transaction using said updated large object header (See column 6, lines 24-36).

8. Regarding claim 3, **Bruso et al.** additionally teaches the sending of an exception to said transaction comprises: sending an error code to a user read thread that generated said transaction. (See column 11, lines 36-38 "If the lock is denied due to a timeout, deadlock, or other reason, step 537 returns an error to the caller").

9. Regarding claim 4, **Bruso et al.** additionally teaches committing said replay of said log entry on the secondary server to delete said smart large object on the secondary server (See column 9, lines 25-28 "FIG 8. is a flowchart of a process of deleting a BLOB value from a row of a database table. The process generally entails finding the storage associated with the BLOB value and returning the storage to a free chain of allocation control table 200"); and invalidating said large object header (See column 9, lines 35-37 "In addition, the step-ID associated with the thread deleting the BLOB value is stored in the BLOB header page that is returned to the free chain.")

10. Regarding claim 5, **Bruso et al.** additionally teaches allocating new space to said smart large object on the primary server, the allocating including: allocating a memory page on the primary server to provide said new space (See column 7, lines 64-65 "...the BLOB handler allocates space in BLOB storage from the BLOB(s)), and subsequent to the allocating of a memory page, updating a header of the smart large object on the primary server (See column 8, lines 1-3 "The operating system...is called by BLOB handler to write the BLOB header page from the BLOB handler's working space to BLOB storage."); wherein the allocating of the memory page and the updating of the header are logged on the primary server to define at least a portion of said log entry log entry that alters said smart large object (See column 8, lines 3-6 "The identifier(s) associated with the BLOB(s) is returned to DBMS 302, and the new row of data, including the identifier(s), is written to database table 304.")

11. Regarding claim 6, **Bruso et al.** additionally teaches deallocating memory from said smart large object on the primary server (See column 9, lines 26-29), the deallocating including: updating a header of the smart large object on the primary server (See column 9, lines 35-37), and subsequent to the updating, deallocating said memory (See column 9, lines 26-29); wherein the updating and deallocating are logged on the primary server to define at least a portion of said log entry that alters said smart large object (See column 9, lines 32-35).

12. Regarding claim 8, **Bruso et al.** additionally teaches the initiating of said transaction comprises: initiating said transaction without locking the smart large object on the secondary server (See column 6, lines 5-6. Here, it specifically notes that locking as an isolation technique is not essential for the invention and suggests prior to these lines that keeping the transactions serializable is another isolation strategy.)

13. Regarding claim 9, **Bruso et al.** teaches a method substantially as claimed. **Bruso et al.** fails to teach performing a modifying transaction on the primary server that alters said smart large object; and logging said modifying transaction on the primary server to generate at least a portion of said log entry that alters said smart large object.

However, **Holenstein et al.** teaches performing a modifying transaction on the primary server that alters said smart large object; and logging said modifying transaction on the

primary server to generate at least a portion of said log entry that alters said smart large object (See page 5, paragraph [0084] "Each node includes a database, an audit trail of all transactions posted to the database, and a transaction transmitter which sends selected transactions in the audit trail to one or more other nodes." The logging as referred to in the claim is performed by the audit trail of the reference.)

It would have been obvious to one with ordinary skill in the art at the time the invention was made to combine the method of **Bruso et al.** with the log entry generation as disclosed in **Holenstein et al.** because the log entry generation keeps track of the changes that will be needed in order to be make the same changes on the replicated databases as well. It is for this reason that one of ordinary skill in the art would have been motivated to include performing a modifying transaction on the primary server that alters said smart large object; and logging said modifying transaction on the primary server to generate at least a portion of said log entry that alters said smart large object.

14. Regarding claim 10, **Bruso et al.** additionally teaches the performing of said modifying transaction comprises: acquiring a lock on said smart large object on the primary server (See column 6, lines 4-6).

15. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Bruso et al.** (6,615,219) in view of **Holenstein et al.** (US 2003/0037029) as applied to claim 1 above, and further in view of **Dinh et al.** (US 2004/0267843). **Bruso et al.** teaches the

initiating of a transaction on the secondary server that includes reading a smart large object comprises: initiating a transaction on the secondary server that includes reading one of a smart binary large object (smart-BLOB) (See column 8, lines 48-53 "If the referenced BLOB is cached, BLOB handler reads the BLOB from the cache and returns the BLOB to Web BLOB server, which returns the BLOB to the requesting application. Otherwise, BLOB handler uses the URL to read the BLOB from BLOB storage.")

Bruso et al. and **Holenstein et al.** fail to teach a smart character large object (smart-CLOB).

However, **Dinh et al.** teaches a smart character large object (smart-CLOB). (See page 3, paragraph [0036] "The use of 'character' as a data type may refer in various embodiments to any type of character data, such as...CLOB (Character Large Object), or any other appropriate type of character data.)

It would have been obvious to one with ordinary skill of the art at that time of the invention to combine the method as disclosed in Bruso et al. and Holentstein et al. with the CLOB of Dinh et al. because both are large objects that are not easily replicated between databases, and the methods for doing so are very similar. Adding CLOBs to the list of data types also does not change the method that is being used, just the type of data that is being replicated. It is for this reason that one of ordinary skill in the art

would have been motivated to also include a smart character large object (smart-CLOB).

16. Claims 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Holenstein et al.** (US 2003/0037029) in view of **Bruso et al.** (6,615,219).

17. Regarding claim 11, **Holenstein et al.** teaches a database including primary and secondary servers and a high availability data replicator adapted to synchronize the secondary server with the primary server (See page 2, paragraph [0033] "...duplicating the contents of at least a portion of data records held in a source database to a target database"), a smart large object application program interface residing on the secondary server (See page 9, paragraph [0173] "Using a 'callable' library application-programming interface (API) that performs the application disk I/O on behalf of the application...")

Holenstein et al. fails to teach the smart large object application program interface comprising: a smart large object read module adapted to read data from a smart large object without acquiring a lock on said smart large object, said read data being communicated to a client; and an exception module adapted to send an exception to said client responsive to a synchronizing event of the high availability data replicator modifying said smart large object.

However, **Bruso et al.** teaches a smart large object read module adapted to read data from a smart large object without acquiring a lock on said smart large object, said read data being communicated to a client (See Column 6, lines 5-7 "Locking as an isolation technique is not essential to the invention. Other equivalent isolation techniques could be employed."); and an exception module adapted to send an exception to said client responsive to a synchronizing event of the high availability data replicator modifying said smart large object (See column 9, lines 19-24 "...if the DBMS detects a processing error which prevents the insert-row operation from being committed, the BLOB pages allocated at step 404 are returned to the appropriate free chain of the allocation control table, along with the BLOB header page." Examiner interprets this as being equivalent to sending an exception.)

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the teaching of **Holenstein et al.** with the read and exception modules as described in **Bruso et al.** because the benefit of not locking the large object allows for the replication to occur while still being able to send an exception in the event that the large object was changed. This provides the most flexibility, while still keeping consistency between the databases. It is for this reason that one of ordinary skill in the art would have been motivated to include the smart large object application program interface comprising: a smart large object read module adapted to read data from a smart large object without acquiring a lock on said smart large object, said read data being communicated to a client; and an exception module adapted to

send an exception to said client responsive to a synchronizing event of the high availability data replicator modifying said smart large object.

18. Regarding claim 12, **Holenstein et al.** teaches a database substantially as claimed. **Holenstein et al.** fails to teach an update module adapted to update said smart large object responsive to said synchronizing event.

However, **Bruso et al.** teaches an update module adapted to update said smart large object responsive to said synchronizing event (See column 6, lines 42-46 "If the transaction requests a commit, the copy of the allocation control table entry and the BLOB header page are discarded and the database updates are made permanent as shown by block 244, which represents the new state of the database.")

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the database of **Holentstein et al.** with the update module as described in **Bruso et al.** because it is necessary for databases that are being synchronized to have a method for updating the object that is being changed. It is for this reason that one of ordinary skill in the art would have been motivated to include an update module adapted to update said smart large object responsive to said synchronizing event.

19. Regarding claim 13, **Holenstein et al.** teaches a database substantially as claimed. **Holenstein et al.** fails to teach a cache module adapted to create a memory cache of said smart large object, said smart large object read module accessing said memory cache during said read, said cache module being further adapted to update said memory cache responsive to said updating of said large smart object.

However, **Bruso et al.** teaches a cache module adapted to create a memory cache of said smart large object (See column 8, lines 62-64 "At step 402, DBMS allocates storage for the row of data to insert, and BLOB storage is allocated at step 404."), said smart large object read module accessing said memory cache during said read (See column 7, lines 45-49 "For operations where a BLOB is to be returned to application program 310 (e.g., a SELECT), the referenced BLOB is read from contiguous storage of BLOB storage and written to a memory location provided by transaction 310"), said cache module being further adapted to update said memory cache responsive to said updating of said large smart object (See column 7, lines 58-62).

It would have been obvious to one having ordinary skill in the art at the time of the invention to include the cache module and read module as disclosed in **Bruso et al.** because the synchronization between the databases using a cache allows the database to be updated on the fly and provides for greater efficiency while still keeping the databases consistent. It is for this reason that one of ordinary skill in the art would have been motivated to include a cache module adapted to create a memory cache of said

smart large object, said smart large object read module accessing said memory cache during said read, said cache module being further adapted to update said memory cache responsive to said updating of said large smart object.

20. Claims 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Holenstein et al.** (US 2003/0037029) as applied to claim 14 above, and further in view of **Bruso et al.** (6,615,219).

21. Regarding claim 15, **Holenstein et al.** teaches an article of manufacture substantially as claimed. **Holenstein et al.** fails to teach identifying a read operation accessing said smart large object; and prior to the replaying of said log entries of said smart large object modifying operation, communicating an exception to a client associated with said read operation. However, **Bruso et al.** teaches identifying a read operation accessing said smart large object (See column 8, lines 48-53 "If the referenced BLOB is cached, BLOB handler reads the BLOB from the cache and returns the BLOB to Web BLOB server, which returns the BLOB to the requesting application. Otherwise, BLOB handler uses the URL to read the BLOB from BLOB storage."); and prior to the replaying of said log entries of said smart large object modifying operation, communicating an exception to a client associated with said read operation (See column 9, lines 19-24 "...if the DBMS detects a processing error which prevents the insert-row operation from being committed, the BLOB pages allocated at step 404 are

returned to the appropriate free chain of the allocation control table, along with the BLOB header page.")

It would have been obvious to one with ordinary skill in the art to combine the teachings of **Holenstein et al.** with the accessing and exception operations of **Bruso et al.** because **Bruso et al.** provides the handling methods for data replication that **Holenstein et al.** does not, and this way of handling the large objects is easily used with the replication method of **Holenstein et al.** It is for this reason that one of ordinary skill in the art would have been motivated to include identifying a read operation accessing said smart large object; and prior to the replaying of said log entries of said smart large object modifying operation, communicating an exception to a client associated with said read operation.

22. Regarding claim 16, **Holenstein et al.** teaches an article of manufacture substantially as claimed. **Holenstein et al.** fails to teach: the replaying of said log entries of said smart large object modifying operation includes modifying said smart large object on the secondary server; and the communicating of an exception includes communicating an error code to said client. However, **Bruso et al.** teaches the replaying of said log entries of said smart large object modifying operation includes modifying said smart large object on the secondary server (See column 8, lines 48-53 "If the referenced BLOB is cached, BLOB handler reads the BLOB from the cache and returns the BLOB to Web BLOB server, which returns the BLOB to the requesting

application. Otherwise, BLOB handler uses the URL to read the BLOB from BLOB storage." And see column 9, lines 25-29 "Fig 8. is a flowchart of a process for deleting a BLOB value from a row of a database table. The process generally entails finding the storage associated with the BLOB value and returning the storage to a free chain of allocation control table 200." This is an example of a modifying operation on a smart large object on the secondary server.); and the communicating of an exception includes communicating an error code to said client (See column 9, lines 19-24 "...if the DBMS detects a processing error which prevents the insert-row operation from being committed, the BLOB pages allocated at step 404 are returned to the appropriate free chain of the allocation control table, along with the BLOB header page." Examiner interprets this as being equivalent to sending an exception.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Holenstein et al.** with the modifying and exception communication of **Bruso et al.** because in order for the synchronizing to be effective, there must be a method of handling modifications of the large object and **Bruso et al.** teaches one that integrates well for the stated purposes. It is for this reason that one of ordinary skill in the art would have been motivated to include the replaying of said log entries of said smart large object modifying operation includes modifying said smart large object on the secondary server; and the communicating of an exception includes communicating an error code to said client.

23. Regarding claim 17, **Holenstein et al.** teaches an article of manufacture substantially as claimed. **Holenstein et al.** fails to teach the replaying of said log entries of said smart large object modifying operation includes deleting said smart large object on the secondary server; and the communicating of an exception includes invalidating a large object header associated with said read operation.

However, **Bruso et al.** teaches the replaying of said log entries of said smart large object modifying operation includes deleting said smart large object on the secondary server (See column 9, lines 25-28 "FIG 8. is a flowchart of a process of deleting a BLOB value from a row of a database table. The process generally entails finding the storage associated with the BLOB value and returning the storage to a free chain of allocation control table 200"); and the communicating of an exception includes invalidating a large object header associated with said read operation (See column 9, lines 35-37 "In addition, the step-ID associated with the thread deleting the BLOB value is stored in the BLOB header page that is returned to the free chain.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Holenstein et al.** with the large object deletion and exception processing of **Bruso et al.** because **Bruso et al.** presents a method that deals with deleting the large object from the server which integrates efficiently into the article of manufacture for replication as disclosed in **Holenstein et al**. It is for this reason that one of ordinary skill in the art would have been motivated to include the

replaying of said log entries of said smart large object modifying operation includes deleting said smart large object on the secondary server; and the communicating of an exception includes invalidating a large object header associated with said read operation.

24. Regarding claim 18, **Holenstein et al.** teaches an article of manufacture substantially as claimed. **Holenstein et al.** fails to teach said smart large object modifying operation includes allocating memory to said smart large object, and said ordering comprises: ordering a log entry corresponding to allocation of memory before a log entry corresponding to updating a header of the smart large object, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order allocates memory on the secondary server before updating said header on the secondary server.

However, **Bruso et al**. teaches said smart large object modifying operation includes allocating memory to said smart large object (See column 7, lines 64-65 "...the BLOB handler allocates space in BLOB storage from the BLOB(s)), and said ordering comprises: ordering a log entry corresponding to allocation of memory before a log entry corresponding to updating a header of the smart large object (See column 8, lines 1-3 "The operating system...is called by BLOB handler to write the BLOB header page from the BLOB handler's working space to BLOB storage."); whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the

selected order allocates memory on the secondary server before updating said header on the secondary server (See column 8, lines 3-6 "The identifier(s) associated with the BLOB(s) is returned to DBMS 302, and the new row of data, including the identifier(s), is written to database table 304.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Holenstein et al.** with the allocating and header updating processes of **Bruso et al.** because it is necessary to handle memory allocation when dealing with database replication, and **Bruso et al.** provides a method for doing so, while handling the large objects which are not well addressed in **Holenstein et al.** It is for this reason that one of ordinary skill in the art would have been motivated to include said smart large object modifying operation includes allocating memory to said smart large object, and said ordering comprises: ordering a log entry corresponding to allocation of memory before a log entry corresponding to updating a header of the smart large object, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order allocates memory on the secondary server before updating said header on the secondary server.

25. Regarding claim 19, **Holenstein et al.** teaches an article of manufacture substantially as claimed. **Holenstein et al.** fails to teach said smart large object modifying operation includes deallocating memory from said smart large object, and said ordering comprises: ordering a log entry corresponding to updating a header of the

smart large object before a log entry corresponding to deallocation of memory, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order updates said header on the secondary server before deallocating memory on the secondary server.

However **Bruso et al.** teaches said smart large object modifying operation includes deallocating memory from said smart large object (See column 9, lines 26-29 "The process generally entails finding the storage associated with the BLOB value and returning the storage to a free chain of allocation control table 200." In other words, deallocating the memory.), and said ordering comprises: ordering a log entry corresponding to updating a header of the smart large object before a log entry corresponding to deallocation of memory (See column 9, lines 35-37 "In addition, the step-ID associated with the thread deleting the BLOB value is stored in the BLOB header page that is returned to the free chain."), whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order updates said header on the secondary server before deallocating memory on the secondary server (See column 9, lines 32-35 "The appropriate free chain the allocation control table 200 is updated at step 454 to include the BLOB pages from the deleted BLOB value.")

It would have been obvious to one with ordinary skill in the art to combine the teachings of **Holenstein et al.** with the deallocating teachings of **Bruso et al.** because a method

is necessary for handling the deallocation of the large objects that is lacking in **Holenstein et al.** and **Bruso et al.** teaches one that efficiently integrates into the apparatus. It is for this reason that one of ordinary skill in the art would have been motivated to include modifying operation includes deallocating memory from said smart large object, and said ordering comprises: ordering a log entry corresponding to updating a header of the smart large object before a log entry corresponding to deallocation of memory, whereby replaying of said log entries of said smart large object modifying operation on the secondary server in the selected order updates said header on the secondary server before deallocating memory on the secondary server.

26. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Holenstein et al.** (US 2003/0037029) as applied to claim 14 above, and further in view of **Dinh et al.** (US 2004/0267843). **Holenstein et al.** teaches said ordering of log entries of a smart large object modifying operation comprises: ordering log entries of one of: a smart binary large object (smart BLOB) modifying operation (See page 5, paragraph [0084] "...and a transaction transmitter which sends selected transactions in the audit trail to one or more other nodes." These transactions are modifying operations performed on the primary server....), performed on the primary server in a selected order wherein a log entry corresponding to updating a large object header of said smart large object is consistent immediately upon execution (See page 5, paragraph [0087] "The marker is written to the audit trail at the first node." Here, the "audit trail" includes what is called the "header" from the claim.)

Holenstein et al. fails to teach a smart character large object (smart-CLOB modifying operation.

However, **Dinh et al.** teaches a smart character large object (smart-CLOB) modifying operation. (See page 3, paragraph [0036] "The use of 'character' as a data type may refer in various embodiments to any type of character data, such as...CLOB (Character Large Object), or any other appropriate type of character data.)

It would have been obvious to one with ordinary skill of the art at that time of the invention to combine the method as disclosed in **Holentstein et al.** with the CLOB of **Dinh et al.** because both are large objects that are not easily replicated between databases, and the methods for doing so are very similar. Adding CLOBs to the list of data types also does not change the method that is being used, just the type of data that is being replicated. It is for this reason that one of ordinary skill in the art would have been motivated to also include a smart character large object (smart-CLOB) modifying operation.

Conclusion

27. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Earl et al. teaches some of the database replication elements and some BLOB elements.

Lamburt teaches database transfer between databases of BLOBs, but a log is not used.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 8:30-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv
12 June 2006

Primary Examiner
Art Unit 2167